Assignment - 2
Multimedia Technology in Mobile Networks
CS6317

Task: To create a iPhone Application that uses Core Data to create, read, update and delete data from the application.

Application Summary: The Application is about a list of Cities in Europe along with its information and images that can be viewed, added, updated, deleted.

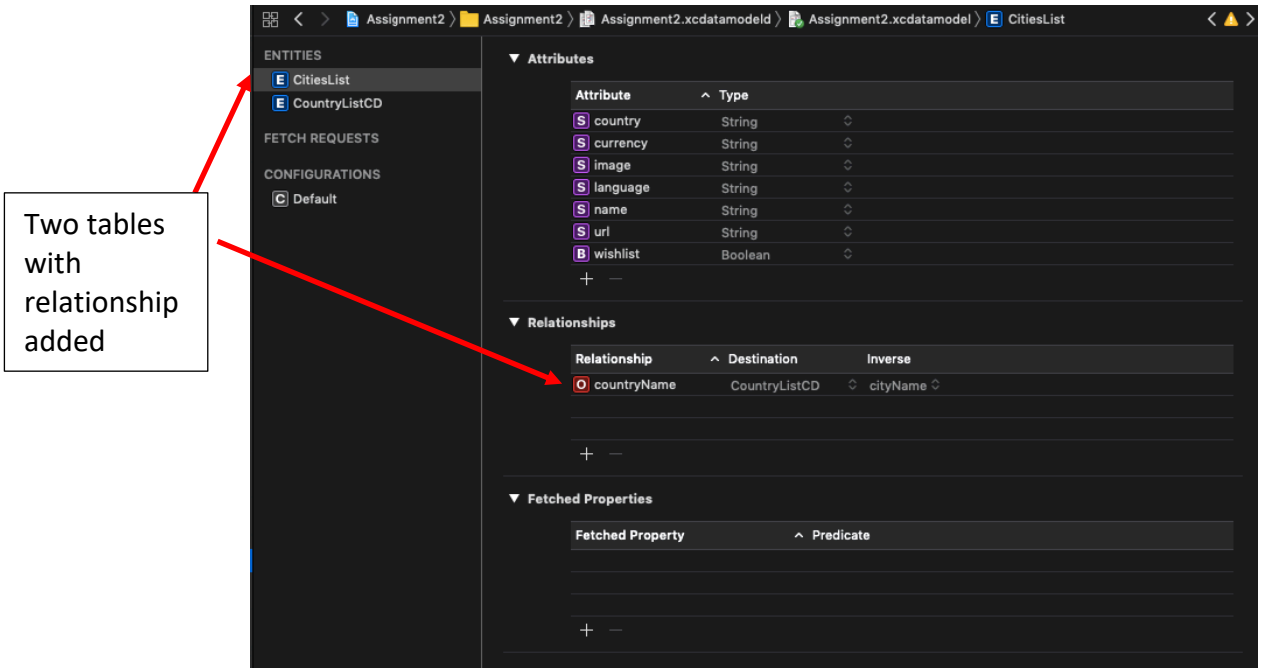
Platform Used: iOS
IDE Used: Xcode
Languages Used: Swift

Technical Information:
The Application begins with a table displaying data fetched from XML about the best cities in Europe. Controls on each screen can help user navigate through the application.

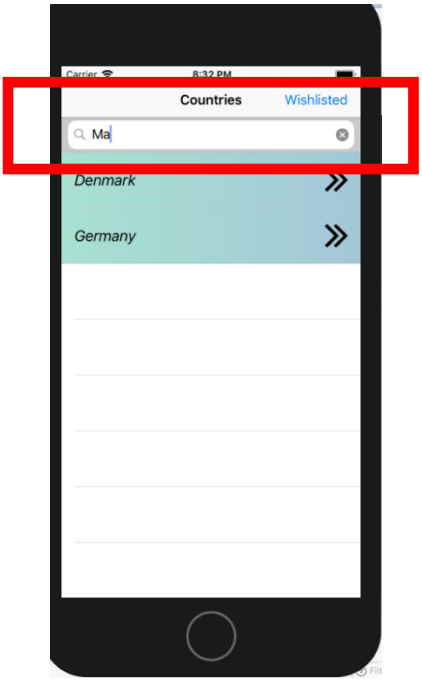
Technical Contributions per screen are mentioned below on an iPhone 8 Simulator:

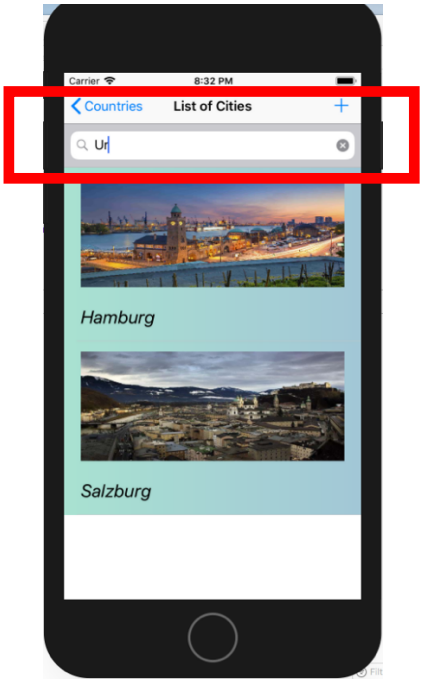
1. **Launch Screen -** The application begins with a launch screen

2. Use of two Tables are used namely CountryList and CityList. Data for each tables is being populated by the XML

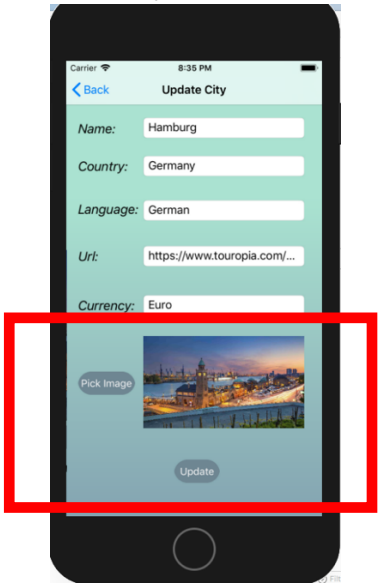


3. Search Bar for tables using predicates instead of the traditional array filter options.

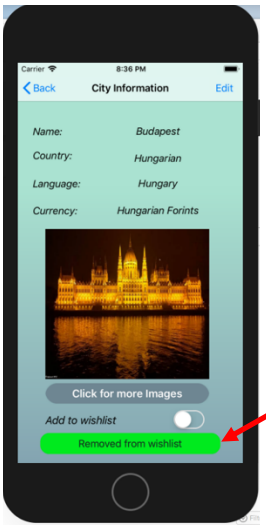




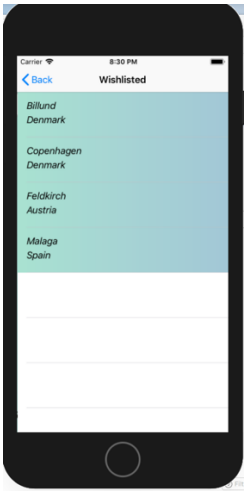
4. Use for Image Picker to add new images and update existing images to Document Directory



5. Use of toasts to display messages



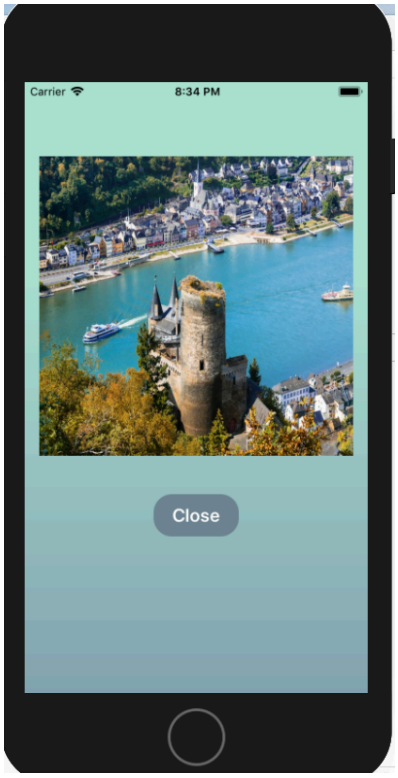
6. Use of predicate to filter data from tables
   Predicate used to filter cities that are added to the wishlist



7. Use of web data using url session

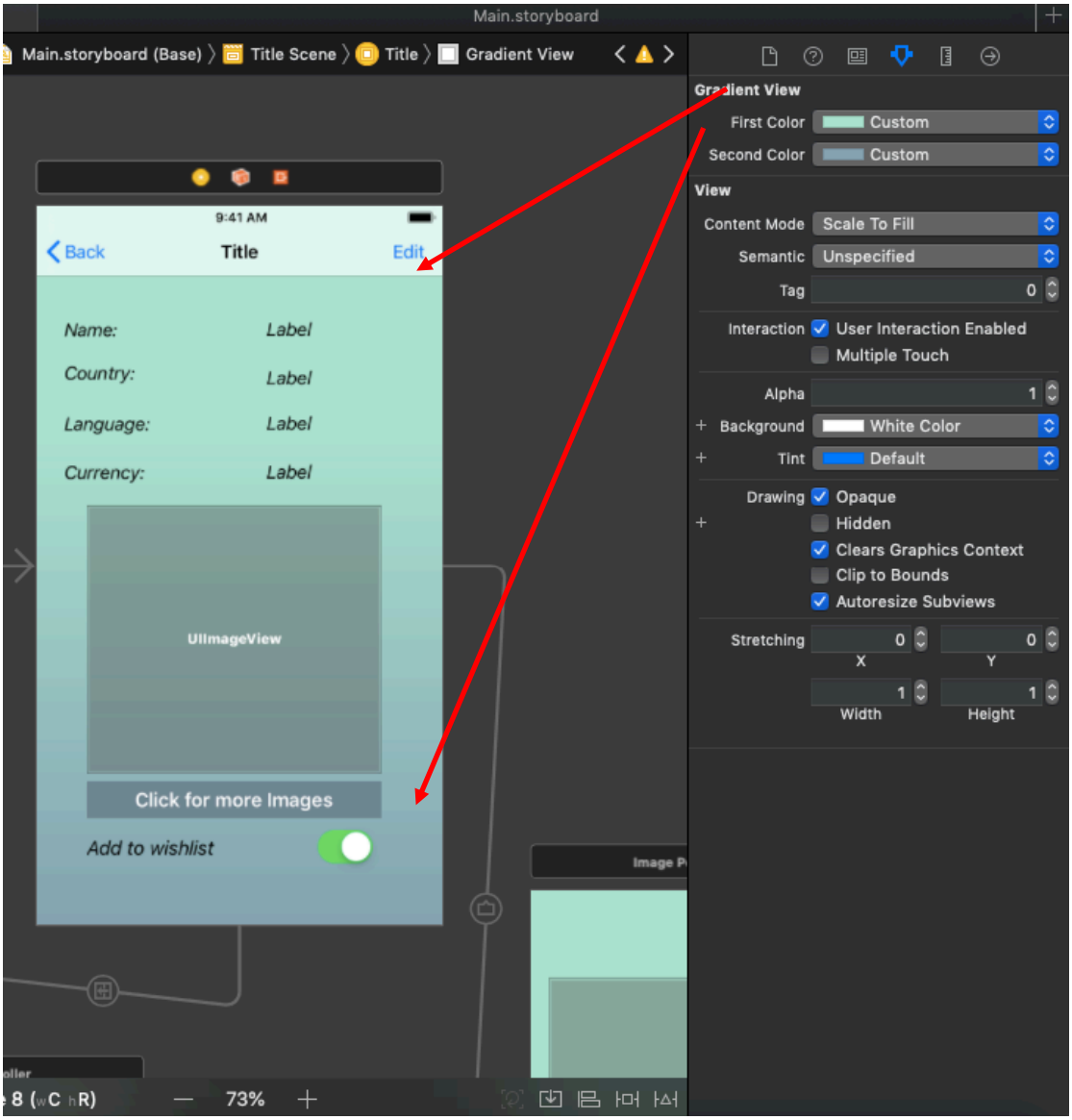
```
@IBOutlet weak var imageView: UIImageView!
override func viewDidLoad() {
    super.viewDidLoad()
    closeButton.layer.cornerRadius = 20
    showActivityIndicator(show: true)
    let sharedSession = URLSession.shared
    if let url = URL(string: url) {
        // Create Request
        if (url != nil){                    ⚠ Comparing non-optional value of type 'URL' to nil always returns true
            let request = URLRequest(url: url)
            let dataTask = sharedSession.dataTask(with: request, completionHandler: { (data, response,
                error) -> Void in
                if let data = data, let image = UIImage(data: data) {
                    DispatchQueue.main.async {
                        self.imageView.image = image
                        self.showActivityIndicator(show: false)
                    }
                }
            })
            dataTask.resume()
        }
            self.imageView.image = UIImage(named: "defaultImage")
            showActivityIndicator(show: false)
        }
    }
```

8. Use of Popover

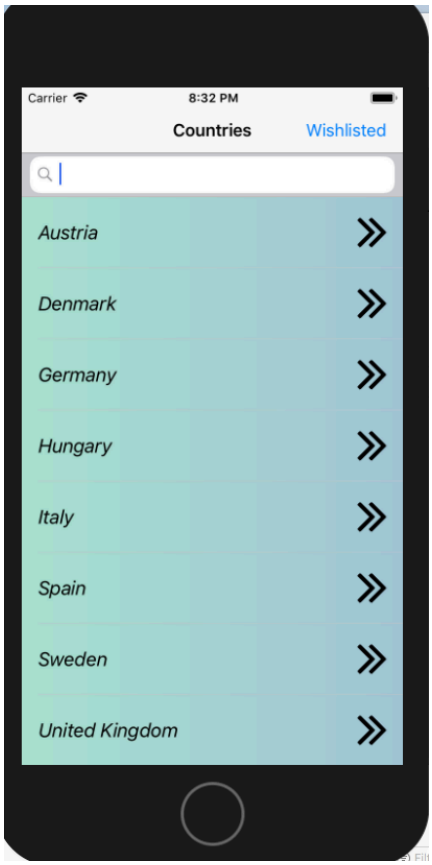


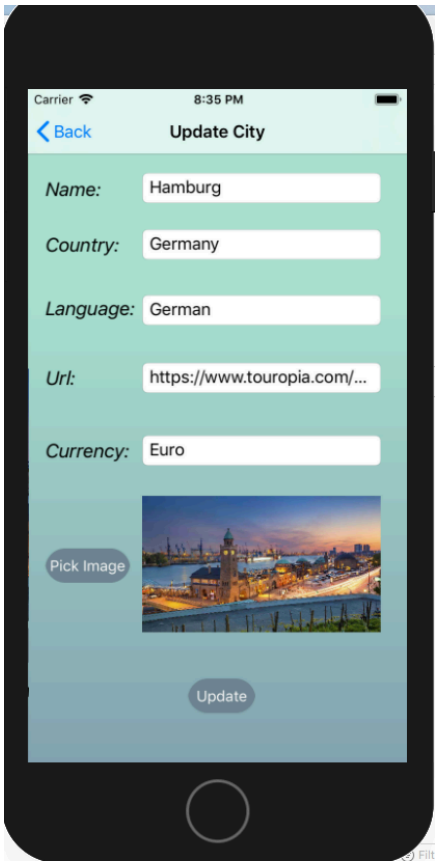
Design Contributions:

1. Gradient Background color used for table cells as well as UIView with provision to select color from storyboard



2. Custom Buttons with applied CSS for the styling purpose.





3. Activity indicator shown until image shows from the Web, but the App still works fine meanwhile. The process is asynchronous